SymmSketch: creating symmetric 3D free-form shapes from 2D sketches

Miao, Yongwei ; Hu, Feixia ; Zhang, Xudong ; Chen, Jiazhou ; Pajarola, Renato

Abstract: This paper presents SymmSketch — a system for creating symmetric 3D free-form shapes from 2D sketches. The reconstruction task usually separates a 3D symmetric shape into two types of shape components, that is, the self-symmetric shape component and the mutual-symmetric shape components. Each type of them can be created in an intuitive manner. According to a uniform symmetry plane, the user first draws 2D sketch lines for each shape component on a sketching plane. The z- depth information of the hand-drawn input sketches can be calculated using their property of mirror symmetry to generate 3D constructive curves. In order to provide more freedom for controlling the local geometric features of the reconstructed free- form shapes (such as their cross sections will not be limited to be traditional circular), our modeling system will create each shape component from four constructive curves. With one pair of symmetric curves and one pair of general curves, an improved cross-sectional surface blending scheme is applied to generate a parametric surface for each component. The final symmetric free- form shape will be progressively created and be represented as 3D triangular mesh. Experimental results illustrate that our system can generate symmetric complex free-form shapes effectively and conveniently.

DOI: https://doi.org/10.1007/s41095-015-0002-8

Originally published at:
Miao, Yongwei; Hu, Feixia; Zhang, Xudong; Chen, Jiazhou; Pajarola, Renato (2015). SymmSketch: creating symmetric 3D free-form shapes from 2D sketches. Computational Visual Media, 1(1):3-16.
DOI: https://doi.org/10.1007/s41095-015-0002-8

SymmSketch: Creating Symmetric 3D Free-form Shapes from 2D Sketches

Yongwei Miao, Feixia Hu, Xudong Zhang, Jiazhou Chen, and Renato Pajarola, *Member, IEEE*

***Abstract*—This paper presents SymmSketch — a system for creating symmetric 3D free-form shapes from 2D sketches. The reconstruction task usually separates a 3D symmetric shape into two types of shape components, that is, the self-symmetric shape component and the mutual-symmetric shape components. Each type of them can be created in an intuitive manner. According to a uniform symmetry plane, the user first draws 2D sketch lines for each shape component on a sketching plane. The z-depth information of the hand-drawn input sketches can be calculated using their property of mirror symmetry to generate 3D constructive curves. In order to provide more freedom for controlling the local geometric features of the reconstructed free-form shapes (such as their cross sections will not be limited to be traditional circular), our modeling system will create each shape component from four constructive curves. With one pair of symmetric curves and one pair of general curves, an improved cross-sectional surface blending scheme is applied to generate a parametric surface for each component. The final symmetric free-form shape will be progressively created and be represented as 3D triangular mesh. Experimental results illustrate that our system can generate symmetric complex free-form shapes effectively and conveniently.**

***Index Terms*—3D reconstruction, constructive curves, free-form shapes, mirror symmetry, sketch-based modeling.**

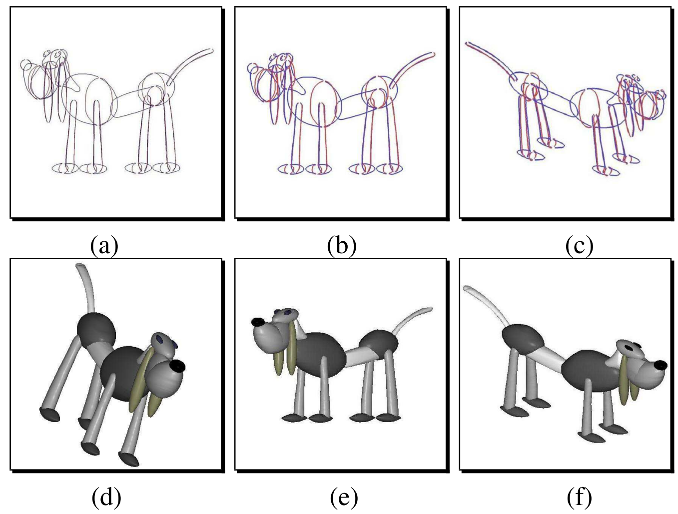


Fig. 1. *Example of the complex symmetric dog model created by our SymmSketch system. (a) shows the user input hand-drawn 2D sketches. (b) and (c) are the corresponding 3D constructive curves generated by our z-depth computation theory of symmetric sketches which are shown in two different view projections. (d), (e) and (f) show the final created 3D free-form shapes in three different view directions respectively.*

I. Introduction

In the area of computer graphics and digital entertainment, sketching, which has been proved to be an important art genre in the early community [1], is now still a common way to convey ideas quickly. Sketch-based modeling has become a popular research topic in the creation of 3D models due to its natural and straightforward manner to access real world objects [2]–[4].

Many studies have shown that inferring a free-form shape from line drawings is very difficult due to its lack of inherent human perception [5]. So far, many sketch-based modeling systems have been proposed, such as SKETCH [6], Teddy [7], ShapeShop [8], SmoothSketch [9], FiberMesh [10], ILoveSketch [11], Rigmesh [12], Sketch2Scene [13] and ArtiSketch [14]. However, many shape modeling operations of these sketch-based systems are always performed in 3D object space. A rough 3D model is first created, and some new features can then be added interactively in a progressive way using some special operations, such as extrusion, cutting, rotation, merging and deformation, etc.

From the viewpoint of practical applications, sketch-based modeling provides a very popular manner for interactively generating 3D shapes [2]. It can offer the user a simple way to access and interpret 3D objects, and thus can effectively avoid the tedious process of operating professional 3D modeling software. Moreover, due to the symmetric property of many real-world objects, it is significant to provide the user a sketch-based reconstruction system for 3D symmetric shapes [15]–[18]. The traditional techniques for modeling symmetric objects from two construction lines will always be limited to the mirror-symmetric shapes with some circular cross sections [19], [20]. In order to provide more freedom for controlling the local geometric features of the reconstructed free-form shapes (such as their cross sections will not be limited to be circular), we present SymmSketch — a novel system for creating symmetric complex 3D shapes from 2D sketches. Using the symmetry information of the 2D input sketches (see Fig. 1a), the 3D symmetric constructive curves can first be computed. The 3D non-symmetric general constructive curves can thus be calculated from the symmetric ones (see Fig. 1b and Fig. 1c). Each shape component can be generated from a pair of symmetric curves and a pair of general curves, and

Manuscript received September 12, 2014; revised December 1, 2014. This research work is supported by the National Natural Science Foundation of China under Grant 61272309 and Grant 61303138. The 3D models are courtesy of the Aim@Shape, Stanford University and Princeton University.

Y. Miao, F. Hu, X. Zhang, J. Chen are with the College of Computer Science and Technology, Zhejiang University of Technology, 310023, China. The corresponding author is Prof. Yongwei Miao(email: ywmiao@zjut.edu.cn; yongweimiao@gmail.com).

R. Pajarola is with the Department of Informatics, University of Zürich, Zürich CH-8050, Switzerland (email: pajarola@ifi.uzh.ch).

the complex symmetric 3D free-form shapes can finally be created (see Fig. 1d, Fig. 1e, and Fig. 1f).

The main contributions of our work can be summarized as follows:

- A progressive method to create symmetric 3D free-form shapes that consist of two types of shape components, i.e. the self-symmetric shape component and the shape component that is symmetric to another one with respect to a symmetry plane.
- A computational theory to recover z-depth information of 3D constructive curves which combine a pair of symmetric constructive curves and a pair of non-symmetric (general) constructive curves.
- An improved cross-sectional surface blending scheme to generate each shape component of symmetric 3D shapes whose cross sections will not be limited to be circular.

The rest of the paper is organized as follows. The related work is briefly reviewed in Section 2. Section 3 explains the z-depth computation theory for generating 3D constructive curves. An elaborate description of our free-form objects modeling system is given in Section 4. Section 5 shows some experimental results and comparisons with the existing methods. Finally, Section 6 concludes the paper and presents some future work.

II. Related Work

To create 3D free-form shapes, designers tend to directly express their design ideas in 2D sketches, and the system should correctly interpret the input sketches to generate the final 3D shapes [21], [22]. Based on a single input of 2D line drawing, many 3D object reconstruction algorithms have been proposed. Here, we only review previous work concerning the sketching-based modeling approaches for 3D free-from shapes. The readers may refer to the surveys [2] and [5] and the references therein for other related techniques.

Many researches attempt to focus on 3D free-from object reconstruction techniques with a user sketching interface which will draw the visible and hidden 2D sketches of the underlying shape [6], [7]. Given an interactive input of 2D free-form strokes, the Teddy system [7] provides the user a sketching interface for easily designing free-form objects and constructing the plausible 3D polygonal meshes. Due to the sketched 2D outlines of 3D objects, Schmidt et al. [8] developed a ShapeShop system that can create the solid models using hierarchical implicit volume models — BlobTrees. Karpenko and Hughes [9] proposed SmoothSketch — a system for inferring plausible complex free-form shapes from a wide class of visible-contour sketches. The FiberMesh system [10] presented by Nealen et al. is also a freeform shape design system which can reconstruct 3D objects from a collection of input curves. Using a non-linear optimization framework, FiberMesh can automatically generate a smooth surface with sharp creases and darts that are controlled by the user input strokes. Bae et al. [11] presented a 3D curve sketching system ILoveSketch which allows the professional designers to design conceptual 3D curve models in an easy pencil-and-paper way. By unifying the modeling and rigging stages of the 3D character animation pipeline, the RigMesh system [12] provided an easy-to-use interface for generating the complex 3D rig characters. By performing sketch-based co-retrieval and co-placement of 3D relevant models, Xu et al. [13] presented a Sketch2Scene system which can automatically turn the input 2D freehand sketches to semantically valid and well arranged 3D scenes. The ArtiSketch system [14] can reconstruct articulated 3D objects according to several articulated 2D sketches, and novel poses of the 3D model can be generated via manipulating the model skeletons. By employing the Laplacian framework for sketch-based editing, Nealen et al. [23] provided an intuitive interface for the user to edit shape silhouettes or create sharp features on 3D triangular meshes.

However, due to the lack of depth information from the input 2D sketches, the object reconstruction process is non-deterministic and the final generated 3D object cannot be unique [24]. To overcome these difficulties, a large number of recent solutions add appropriate constraints to the original sketches [19], [25] or match sketches to existing 3D models using an evocative system [26]–[28]. Li et al. [29] proposed a modeling system for generating the piecewise planar 3D objects from the object edges drawn on the input image. Using the optimized-based computation of 3D information, Wang et al. [30] developed an approach for creating curved objects from single 2D line drawings. Xue et al. [31] presented a 3D modeling approach for recovering regular geometry from a single image of a symmetric object. Chen et al. [32] also introduced an interactive 3-sweep technique for extracting the simple man-made editable objects from an input image. The 3D shapes reconstructed by these methods are comparatively simple and always be restricted to some specific kinds of objects. Andre et al. [20] presented a simple 3D reconstruction scheme in which each object part can be constructed from two construction lines. However, much user interaction is needed to reconstruct complex objects correctly because their method has no information of the relative depths among multiple parts.

Without extra operations to assemble different shape components, Cordier et al. [19] exploited the mirror symmetry property to create 3D models. Actually, the symmetry assumption is one of the least restrictions for 2D sketches due to the fact that many real-world 3D objects exhibit certain degrees of symmetry [15]–[17], such as animals, buildings, and many other organic structures. Under the orthographic or perspective projection, the 3D structure of a shape with certain degrees of symmetry can fully be reconstructed as long as one can determine the symmetric pairs [18]. According to the hand-drawn input sketches of bilaterally symmetric objects, Oztireli et al. [33] presented a 3D modeling algorithm from a set of planar curves. Using a set of mirror-symmetric curves, Cordier et al. [34] also introduced a 3D reconstruction method for generating some particular types of wire-frame objects.

The aim of our modeling system is to focus on some natural modeling techniques to generate symmetric complex free-form objects from the user input sketches. The traditional techniques for modeling symmetric objects will always be limited to generate simple 3D shapes with some circular cross sections due to their reconstruction scheme from only two construction lines [19]. Compared with the traditional modeling systems, the most important difference of our SymmSketch system is that each shape component can be generated from four constructive curves — a pair of symmetric curves and a pair of non-symmetric general curves which will provide more freedom to control the local geometric features of the final 3D free-form shapes, such as their cross sections will not be limited to be circular.

III. Z-depth Computation for Constructive Curves

A. *Definitions and assumptions*

To clarify the use of hand-drawn input sketches for generating 3D constructive curves, some definitions commonly used in practical applications are given and some assumptions are also presented to simplify the 3D reconstruction process. The reader can refer to Fig. 2.

The **Sketching Plane** is a plane on which the user can draw 2D sketch lines interactively. Without loss of generality, the sketching plane can be selected as the XOY plane $z = 0$. The user input 2D sketches on the sketching plane can thus be considered as the orthogonal projection of the constructive curves of a 3D shape. In our modeling system, only one sketching plane is adopted because it is difficult and inconvenient to draw several overlapped strokes projected on different planes from different components.

The **Symmetry Plane** is a special plane with respect to which the mirror-symmetric shape components exhibit their property of symmetry. The symmetric free-form shapes will be invariant under reflection transformations with regard to their symmetry plane. Our modeling system is designed to use only a unique symmetry plane for reconstructing the whole 3D shape.

Symmetric Curves are the user input sketches whose corresponding 3D constructive curves will be symmetric with respect to the symmetry plane. Each point sampled on these sketches curves will be called a **Symmetric Point**. Here, the property of symmetry is maintained for the curves in 3D space rather than the projection ones on the sketching plane. It is because a curve which is mirror-symmetric in 3D space may not be mirror-symmetric if they are projected to 2D plane.

General Curves are the user input non-symmetric general sketches for representing shape components. Each point sampled on these general curves will be called a **General Point**.

Constructive Curves are spatial curves in 3D space that are recovered from the symmetric or general input sketches using our z-depth computation theory. We assume each component of the reconstructed complex shapes will be generated from four constructive curves, i.e., two symmetric and two non-symmetric general constructive curves.

To effectively reconstruct complex free-form shapes, different from traditional methods [19], [20], our modeling system manages to generate each component of free-form objects by taking four sketches on the sketching plane as input. When the user draws the 2D sketches on the sketching plane, each sketch curve will be uniformly sampled with equal number of sample points and lifted up to a 3D constructive curve after determining their z-coordinates of these sample points. For

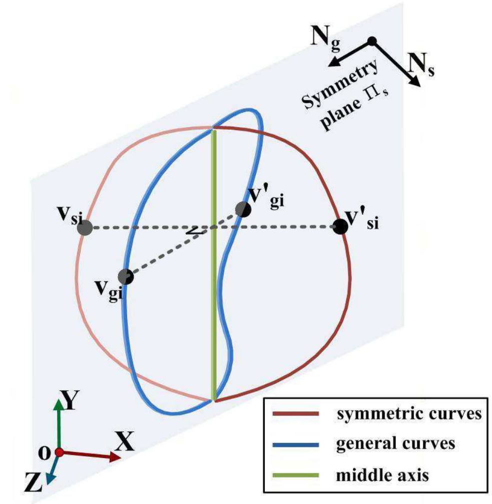


Fig. 2. *Z-depth computation for constructive curves of one self-symmetric shape component.*

reconstructing complex 3D free-form shapes, the four corresponding constructive curves will be processed simultaneously, and the z-depth information of the constructive curves can be calculated using their property of mirror symmetry. That is, the two corresponding recovered symmetry points of two constructive symmetric curves should be mirror-symmetric with respect to the symmetry plane, whilst the vector connecting the two corresponding recovered symmetric points always be perpendicular to the vector connecting the two corresponding recovered general points.

For the sake of simplicity, the input sketches can be considered as the orthogonal projection of 3D constructive curves onto the XOY sketching plane. The x and y coordinates of vertices on the input sketches can simply be taken as the coordinates of the corresponding reconstructed 3D vertices (see Fig. 2). Thus the main remaining problem of 3D object reconstruction is the estimation of the z coordinates of the object vertices.

B. *Z-depth computation theory*

In order to create symmetric complex free-form objects, we suppose that each 3D shape can be separated into two types of components, one is a self-symmetric shape component and the other is the mutual-symmetric shape components with respect to a unique predefined symmetry plane. That is, for each point on the surface of self-symmetric component, there is a symmetric corresponding point locating on the same component. Meanwhile, for each point on one of the mutual-symmetric components, there is a symmetric point existing on the other symmetric component. According to the user input sketches for these two types of shape components, the z-coordinate information of the constructive curves can be determined by two different schemes using their property of mirror symmetry.

1) Z-depth computation of constructive curves representing one self-symmetric component: Due to the property of mirror symmetry, the z coordinates of the sample points on the symmetric curves and general curves of one self-symmetric shape component can be calculated as follows.

Without loss of generality, the symmetry plane Π_s is simply assumed to pass through the origin of the coordinate system with its normal direction $N_s(x^s, y^s, z^s)$. Let $V_s = \{v_{s_0}, v_{s_1}, \cdots, v_{s_{n-1}}\}$ and $V'_s = \{v'_{s_0}, v'_{s_1}, \cdots, v'_{s_{n-1}}\}$ be two sets of n points sampled from the recovered symmetric sketches respectively. Each sample point $v'_{s_i}(x'_{s_i}, y'_{s_i}, z'_{s_i})$ is the mirror image of $v_{s_i}(x_{s_i}, y_{s_i}, z_{s_i})$ with respect to the symmetry plane Π_s. Let $V_g = \{v_{g_0}, v_{g_1}, \cdots, v_{g_i}, \cdots, v_{g_{n-1}}\}$ and $V'_g = \{v'_{g_0}, v'_{g_1}, \cdots, v'_{g_i}, \cdots, v'_{g_{n-1}}\}$ be another two sets of n points sampled from the recovered general sketches respectively. Each pair of two different points $v_{g_i}(x_{g_i}, y_{g_i}, z_{g_i})$ and $v'_{g_i}(x'_{g_i}, y'_{g_i}, z'_{g_i})$ satisfies the following conditions: 1) The vector $v_{g_i} - v'_{g_i}$ is perpendicular to the normal N_s; and 2) The line determined by v_{g_i} and v'_{g_i} will intersect with the middle axis of the symmetric curves.

Given the symmetry plane Π_s as shown in Fig. 2, we choose one pair of symmetric vertices v_{s_i}, v'_{s_i} on the symmetric curves and one pair of general vertices v_{g_i}, v'_{g_i} on the general curves locating on the symmetry plane Π_s. The dashed lines connecting to v_{s_i} and v'_{s_i}, v_{g_i} and v'_{g_i} will intersect at one point. Thus, the following equations can be obtained for these four points v_{s_i}, v'_{s_i}, v_{g_i} and v'_{g_i},

$$\begin{aligned}(v_{s_i} + v'_{s_i}) \cdot N_s &= 0\\ (v_{s_i} - v'_{s_i}) \cdot N_g &= 0\\ (v_{g_i} - v'_{g_i}) \cdot N_s &= 0\\ (2v_{g_i} - v_{s_i} - v'_{s_i}) \cdot N_s &= 0\end{aligned}$$

where N_g is a vector perpendicular to the vector N_s. Using the coordinates of the vertices, these four equations can thus be represented as follows respectively,

$$(x_{s_i} + x'_{s_i})x^s + (y_{s_i} + y'_{s_i})y^s + (z_{s_i} + z'_{s_i})z^s = 0 \tag{1}$$

$$(x_{s_i} - x'_{s_i})x^g + (y_{s_i} - y'_{s_i})y^g + (z_{s_i} - z'_{s_i})z^g = 0 \tag{2}$$

$$(x_{g_i} - x'_{g_i})x^s + (y_{g_i} - y'_{g_i})y^s + (z_{g_i} - z'_{g_i})z^s = 0 \tag{3}$$

and

$$\begin{aligned}(2x_{g_i} - x_{s_i} - x'_{s_i})x^s + (2y_{g_i} - y_{s_i} - y'_{s_i})y^s\\ + (2z_{g_i} - z_{s_i} - z'_{s_i})z^s = 0\end{aligned} \tag{4}$$

By combining the equalities (1) and (2), the z-coordinates z_{s_i}, z'_{s_i} of the symmetric points v_{s_i}, v'_{s_i} can be calculated as follows,

$$\begin{aligned}z_{s_i} = -\frac{1}{2}\Big(&\frac{(x_{s_i} + x'_{s_i})x^s}{z^s} + \frac{(y_{s_i} + y'_{s_i})y^s}{z^s}\\ &+ \frac{(x_{s_i} - x'_{s_i})x^g}{z^g} + \frac{(y_{s_i} - y'_{s_i})y^g}{z^g}\Big)\end{aligned} \tag{5}$$

$$\begin{aligned}z'_{s_i} = -\frac{1}{2}\Big(&\frac{(x_{s_i} + x'_{s_i})x^s}{z^s} + \frac{(y_{s_i} + y'_{s_i})y^s}{z^s}\\ &- \frac{(x_{s_i} - x'_{s_i})x^g}{z^g} - \frac{(y_{s_i} - y'_{s_i})y^g}{z^g}\Big)\end{aligned} \tag{6}$$

Finally, by combining the equalities (3), (4), (5) and (6), the z-coordinates z_{g_i}, z'_{g_i} of the general points v_{g_i}, v'_{g_i} can be determined as follows,

$$\begin{aligned}z_{g_i} = -\frac{1}{2}\Big(&\frac{(2x_{g_i} - x_{s_i} - x'_{s_i})x^s}{z^s} + \frac{(2y_{g_i} - y_{s_i} - y'_{s_i})y^s}{z^s}\\ &+ \frac{(x_{s_i} + x'_{s_i})x^s}{z^s} + \frac{(y_{s_i} + y'_{s_i})y^s}{z^s}\Big)\end{aligned} \tag{7}$$

$$\begin{aligned}z'_{g_i} = -\frac{1}{2}\Big(&\frac{(2x'_{g_i} - x_{s_i} - x'_{s_i})x^s}{z^s} + \frac{(2y'_{g_i} - y_{s_i} - y'_{s_i})y^s}{z^s}\\ &+ \frac{(x_{s_i} + x'_{s_i})x^s}{z^s} + \frac{(y_{s_i} + y'_{s_i})y^s}{z^s}\Big)\end{aligned} \tag{8}$$

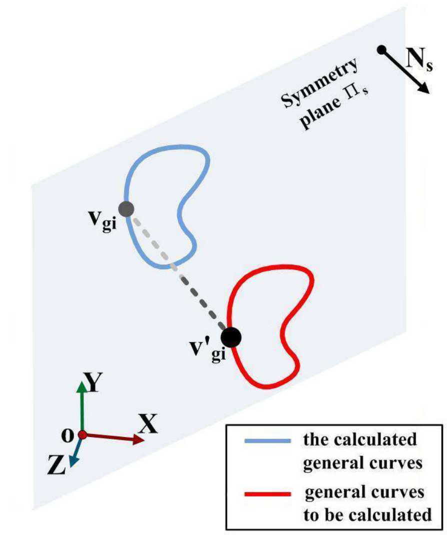


Fig. 3. *Z-depth computation for constructive curves of two mutual-symmetric components of the same object.*

2) Z-depth computation of constructive curves representing two mutual-symmetric components: Now, if one shape component is mirror-symmetric to another component of the same object (see Fig. 3), the z coordinates of the sample points on two pairs of symmetric curves can be easily obtained using Equations (5) and (6). Thus, the z-depth information of the sample points on only one pair of the general curves needs to be computed using Equations (7) and (8), and that of the other pair of general curves can be easily determined using the property of mirror symmetry. For example, for a general point $v_{g_i}(x_{g_i}, y_{g_i}, z_{g_i})$ sampled from the general curves (blue curves) in Fig. 3, its symmetric one $v'_{g_i}(x'_{g_i}, y'_{g_i}, z'_{g_i})$ (the point on the red curves) on the other shape component with respect to the symmetry plane can thus be determined as follows,

$$\begin{aligned}(v_{g_i} - v'_{g_i}) \times N_s &= 0\\ (v_{g_i} + v'_{g_i}) \cdot N_s &= 0\end{aligned}$$

that is,

$$(z_{g_i} - z'_{g_i})x^s - (x_{g_i} - x'_{g_i})z^s = 0 \tag{9}$$

$$(x_{g_i} - x'_{g_i})y^s - (y_{g_i} - y'_{g_i})x^s = 0 \tag{10}$$

$$(x_{g_i} + x'_{g_i})x_s + (y_{g_i} + y'_{g_i})y_s + (z_{g_i} + z'_{g_i})z_s = 0 \tag{11}$$

By combining the equalities (9), (10), and (11), the coordinate of the general point x'_{g_i}, y'_{g_i}, z'_{g_i} can be calculated as follows,

$$x'_{g_i} = \frac{((y^s)^2 + (z^s)^2 - (x^s)^2)x_{g_i} - 2x^s(y_{g_i}y^s + z_{g_i}z^s)}{(x^s)^2 + (y^s)^2 + (z^s)^2} \tag{12}$$

$$y'_{g_i} = \frac{((x^s)^2 + (z^s)^2 - (y^s)^2)y_{g_i} - 2y^s(x_{g_i}x^s + z_{g_i}z^s)}{(x^s)^2 + (y^s)^2 + (z^s)^2} \tag{13}$$

$$z'_{g_i} = \frac{((x^s)^2 + (y^s)^2 - (z^s)^2)z_{g_i} - 2z^s(x_{g_i}x^s + y_{g_i}y^s)}{(x^s)^2 + (y^s)^2 + (z^s)^2} \tag{14}$$

In the following section, the above z-depth computation theory of constructive curves will be adopted by our SymmSketch system to create symmetric complex free-form shapes.

IV. 3D Reconstruction of Symmetric Free-form Shapes

To reconstruct symmetric 3D shapes, Cordier et al. [19] only adopted two symmetric curves which made the final shape consisting of the cylinder-like components, that is, the cross section of each shape component is always circular. Their method can only generate relatively simple and naive 3D shapes due to its limitation to create from only two symmetric curves. The motivation of our work is to provide more freedom for effectively controlling the reconstructed shape component to be enforced some specific geometric features and artistic effect. By adopting a pair of symmetric constructive curves and a pair of general constructive curves, our modeling system can create some complex free-form shapes with relatively flat or sharp component. In this section, we will demonstrate how to classify the user's input sketches and determine the four constructive curves for each shape component. Furthermore, based on the 3D constructive curves, we will also describe how to generate a complex mirror-symmetric 3D free-form shape.

A. Overview of our object reconstruction algorithm

Taking the planar sketches that may consist of symmetric and general curves as input, the 3D coordinates of the sample points can be effectively recovered by using the z-depth computation outlined in Section 3. The final output 3D symmetric free-form shapes will be created and represented as triangular meshes. The high-level framework of our 3D reconstruction algorithm can be summarized as follows:

Step 1. **Discretization of 2D sketching lines**. Once the user draws the 2D sketches on the sketching plane, our modeling system will automatically capture the sketch points of the freehand sketches. The input sketches will then be discretized as polygons whose vertices are uniformly sampled on the smooth quadratic B-spline curves that interpolate these sketch points. Each pair of symmetric or general curves will be stored one by one according to the user's drawing order.

Step 2. **Calculation of 3D constructive curves**. In our modeling system, several groups of 3D curves can be constructed by two types of curve sets, symmetric curves $\prod_s = (c_{s_0}, c'_{s_0}, \cdots, c_{s_{n-1}}, c'_{s_{n-1}})$ (such as the purple curves in Fig. 2) with respect to a unique symmetry plane, and non-symmetric general curves $\prod_g = (c_{g_0}, c'_{g_0}, \cdots, c_{g_{n-1}}, c'_{g_{n-1}})$ (such as the blue curves in Fig. 2). Our reconstruction algorithm will first process all symmetric curves. The 3D coordinate information of these symmetric constructive curves can then be computed directly with respect to the symmetry plane. Meanwhile, the 3D coordinate information of each pair of general constructive curves can be calculated together with the symmetric curves for different types of shape components.

Step 3. **Generation of parametric surfaces by our improved cross-sectional blending scheme**. With one pair of 3D symmetric constructive curves and one pair of 3D general constructive curves, an improved cross-sectional surface blending scheme is applied to generate a parametric surface for each shape component. The final complex free-form shapes will be progressively generated and may consist of several shape components.

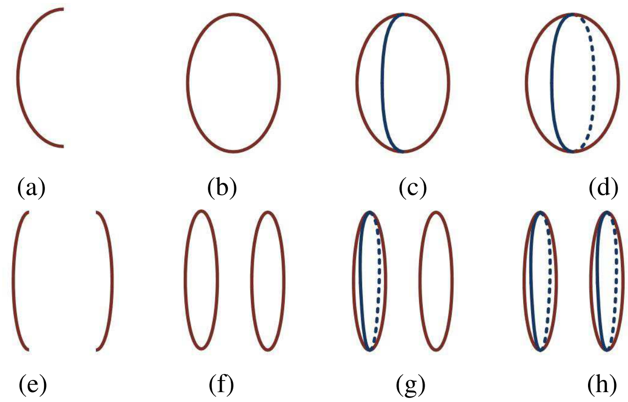


Fig. 4. *Interactively sketching shape components step by step. (a,b,c,d) shows the drawing orders of 2D sketching for self-symmetric shape component, and (e,f,g,h) illustrates the drawing orders of 2D sketching for mutual-symmetric shape component.*

B. Discretization of 2D sketching lines

As is shown in Fig. 2, under the orthogonal projection, user interactively sketches 2D symmetric or general curves for each shape component on the sketching plane. According to Blair's scheme [1] for exhibiting shapes using several profiles, for the self-symmetric component, the user always sketches a pair of symmetric curves and then draws a pair of general curves (see Fig. 4a-Fig. 4d). These four curves usually connect at their endpoints. For two mutual-symmetric shape components, the user will sketch two pairs of symmetric curves for both components and then draw the pairs of general curves for each component respectively (see Fig. 4e-Fig. 4h). Each pair of mutual-symmetric curves on these two components usually have no common connection at their endpoints.

As input, these sketches will be stored with a unique index according to the user's drawing order. Thus, the pair of symmetric or general curves will be handled together. Accordingly, each pair of general curves will be coupled with the corresponding symmetric curves. Furthermore, using the sketch points that automatically sampled from the sketch lines, these freehand sketches will always be approximately represented as polygons with some sketch vertices. These sketch vertices can be uniformly sampled on the quadratic B-spline curves which interpolate the sampled sketch points [35]. For the sake of simplicity, each pair of sketch lines will be sampled by the same number of sample points for z-depth recovery using our computational scheme outlined in next section.

C. Calculation of 3D constructive curves

The input sketch lines will be stored according to the user's drawing order. The system will first classify all of these input sketches into different nodes as $L = \{I_0, I_1, \cdots, I_i, \cdots, I_{n-1}\}$. Every four curves drawn by the user will be stored in the same node. All of the original nodes in L will be divided into two types of nodes $L_p = \{I_{p_0}, I_{p_1}, \cdots, I_{p_i}, \cdots, I_{p_{n-1}}\}$ and $L_q = \{I_{q_0}, I_{q_1}, \cdots, I_{q_i}, I_{q_{i+1}}, \cdots, I_{q_{n-2}}, I_{q_{n-1}}\}$. Each node I_{p_i} in L_p contains the sketch curves of recovering a single self-symmetric shape component, and each pair of symmetric curves always meet at their endpoints (see Fig. 4a-Fig. 4d). The nodes $I_{q_{2j}}$ and $I_{q_{2j+1}}$ in L_q contain the sketch curves of two mutual-symmetric shape components in the same model (for $0 \leq j \leq [\frac{n-1}{2}]$) (see Fig. 4e-Fig. 4h). In general, for reconstructing free-form objects effectively, there must be an even number of nodes stored in L_q. The original node $I_{q_{2j}}$ always contains two pairs of symmetry curves depicting two shape components, whilst the original node $I_{q_{2j+1}}$ always stores two pairs of general curves representing two shape components. Otherwise, the modeling system will warn the user to redraw the sketch curves.

For different types of nodes in L_p and L_q, the system will adopt the different computational schemes to recover their z-depth information of vertices sampled from their sketch curves.

- 1) For creating the self-symmetric shape component from one node $I_{p_i} \in L_p$, the system will first calculate the z coordinates of the symmetric points using Equations (5) and (6). The z coordinates of the general points can then be computed from the 3D symmetric points using Equations (7) and (8).
- 2) For generating two mutual-symmetric shape components from two nodes $I_{q_{2j}}, I_{q_{2j+1}} \in L_q$, node $I_{q_{2j}}$ always contains two pairs of symmetric curves whose corresponding constructive curves will be computed using Equations (5) and (6). The generated symmetric constructive curves will be re-paired so that two pairs of symmetric curves will be re-inserted into $I_{q_{2j}}$ and $I_{q_{2j+1}}$ respectively (see Fig. 5). Each node $I_{q_{2j}}$ and $I_{q_{2j+1}}$ will contain one pair of symmetric constructive curves and one pair of general curves. Thus, the z coordinates of the general points of $I_{q_{2j}}$ can be calculated from the symmetric ones in the same node using Equations (7) and (8). The general curves of $I_{q_{2j+1}}$ will finally be treated as the symmetry image of the general curves of $I_{q_{2j}}$, and their z-depth information will be computed using Equations (12), (13) and (14).

Here, an example for illustrating our procedure of re-pairing and re-inserting different sketches is given in Fig. 5. The head, body and tail part of the doll are self-symmetric which will be represented by four sketch curves in I_{p_i} respectively. For mutual-symmetric ear part, according to the drawing order, the four symmetric curves for depicting two ears of the doll will be stored in one node and the four general curves will be stored in another node at first (see the top right figure of Fig. 5). After our re-pairing step for sketch lines, each pair of symmetric curves will be separated into two nodes respectively, and the symmetric curves of two ears will be included in $I_{q_{2j}}$ and $I_{q_{2j+1}}$ respectively (see the down right figure of Fig. 5).

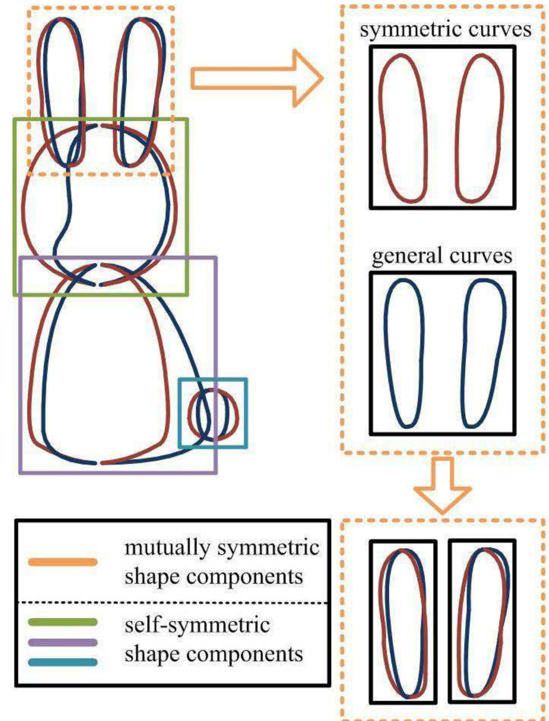
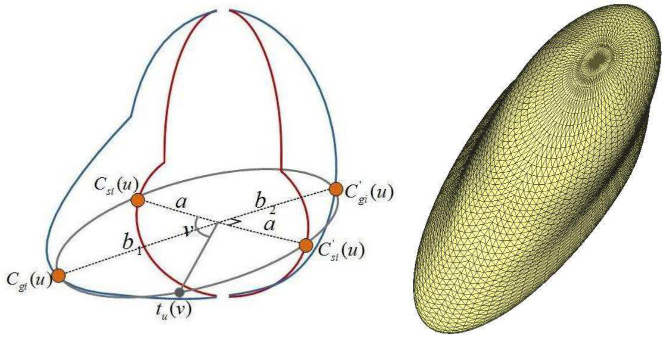


Fig. 5. *An example for re-pairing the input symmetric and general sketches*

As a result, all the 3D coordinate information of the input sketches can be determined correctly. The input 2D sketches will thus be lifted up to 3D constructive curves for subsequent free-form surface generation.

D. Generation of parametric surfaces by our improved cross-sectional blending scheme

For each node I_i, the z-depth information of its sketch curves has been recovered and the 3D constructive curves have been obtained. The next step is to generate a smooth surface to fit these constructive curves of each node. For the sake of reconstructing 3D free-form objects, Severn et al. presented a surface blending scheme [36] to generate the parametric blending surface by sweeping a variable sized circle along a medial axis of two planar sketches. Here, given four 3D constructive curves, our cross-sectional surface blending scheme will sweep a ring of two semi-ellipses so that the final parametric surface can pass through these construction curves (see Fig. 6a). As we have pointed out in Section 3.1 that each of four constructive curves are sampled by the same number of points, every four-points sampled on different curves respectively will be adopted here to generate a planar sweeping ring. However, the selected four-points in order might not locate on the same plane in general. To overcome this issue, our improved cross-sectional blending scheme first determines a plane passing through the center of the line segment connecting two symmetric points, whose normal is determined by the cross product of the two directions connecting the general points and the symmetric points respectively. Then, the two general points can be updated by the intersection points of this plane with two general curves respectively. As a result, the four points



(a) (b)

Fig. 6. *The parametric surface generated for one shape component. Four sample points determine a sweeping ring of two semi-ellipses (a) that are smoothly connected on the symmetric points; (b) shows the final generated shape component.*

can be located in the same plane and can be interpolated by two parameterized semi-ellipses (see Fig. 6a) as follows.

Let $C_{s_i}(u)$, $C'_{s_i}(u)$ be the symmetric curves and $C_{g_i}(u)$, $C'_{g_i}(u)$ be the general curves. For each fixed parameter u, the generated sweeping ring will be parameterized by two semi-ellipses $t_u(v)$ as follows: one semi-ellipse passing through $t_u(0) = C_{s_i}(u)$, $t_u(\frac{\pi}{2}) = C_{g_i}(u)$, $t_u(\pi) = C'_{s_i}(u)$, and the other semi-ellipse passing through $t_u(\pi) = C'_{s_i}(u)$, $t_u(\frac{3\pi}{2}) = C'_{g_i}(u)$, $t_u(2\pi) = C_{s_i}(u)$. Thus the reconstructed parametric surface $S(u, v)$ can be created by translating $t_u(v)$ along the medial axis of $C_{s_i}(u)$ and $C'_{s_i}(u)$. The final shape component can be generated by a parametric blending surface $S(u, v) = t_u(v)$ and be further discretized as triangle meshes (see Fig. 6b).

E. Progressive generating complex symmetric free-form shapes

To create complex symmetric free-form shapes, our presented SymmSketch system will separate whole object into several components. Each shape component will be created one by one, and the final shape can thus be generated in a progressive manner. Once the user finished the sketches for one component, our modeling system will create the corresponding 3D shape component interactively. Our progressive modeling process will provide the user an intuitive design pattern. The user can continue the sketches drawing if they have finished the inputting task for previous shape component. They can also remove some current inaccurate sketches thus for redrawing them. To create some complex shape components, the user can also add some extra sketches for effectively generating some geometric features. Fig. 7 illustrates an example for progressively creating one symmetric 3D free-form shape using our SymmSketch system.

V. Experimental Results and Discussion

All the algorithms presented in this paper have been implemented in C++ with OSG (Open Scene Graph) for graphics display, running on a 2.6 GHz Pentium(R) Dual-Core PC. According to the user input 2D sketches, the main steps

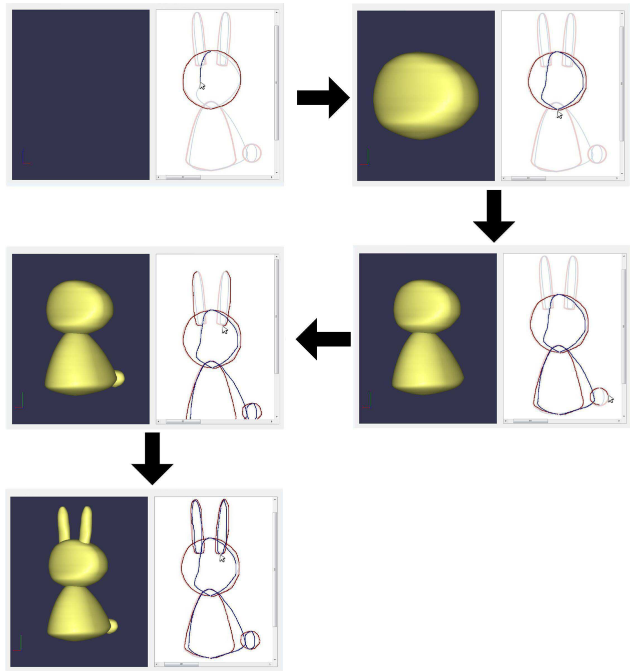


Fig. 7. *Creating complex symmetric 3D shapes in a progressive manner.*

of our reconstruction approach include the computation of 3D constructive curves for each shape component and the generation of smooth parametric surfaces using an improved cross-sectional blending scheme. The experimental results show its effectiveness for creating various types of 3D free-form symmetric shapes in a progressive manner.

A. Creation of symmetric 3D free-form shapes

Our SymmSketch system is suitable for different types of 3D free-form shapes, especially mirror-symmetric complex shapes consisted of several components. Our modeling system has been tested by several novice student users which show its effectiveness and also be interesting for creating 3D symmetric shapes. After about 5 minutes' training on 2D sketching rules, the user can created several common objects and simple cartoon characters interactively (see Fig. 8). For a freehand sketch, the user's sketches may not be very accurate for representing the real symmetric objects. It is worth to emphasize that our reconstruction approach is insensitive to tiny man-made errors of the input hand drawings because our computational theory for z-depth information of constructive curves will intrinsically maintain the property of mirror symmetry.

Moreover, in order to accurately create 3D free-form shapes, the user can insert an existing 2D line drawings to assist sketching tasks. The designers can simply depict the sketch lines on the sketching plane according to the provided line drawings. To create complex 3D free-form objects, our SymmSketch system can effectively control shape components to enforce some special artistic effect, such as the body of a duck, the head of an ostrich and dog, and the plumage of a bird as shown in Fig. 9. Here, the first column (a) in Fig. 9 shows the referenced line drawings on real papers.

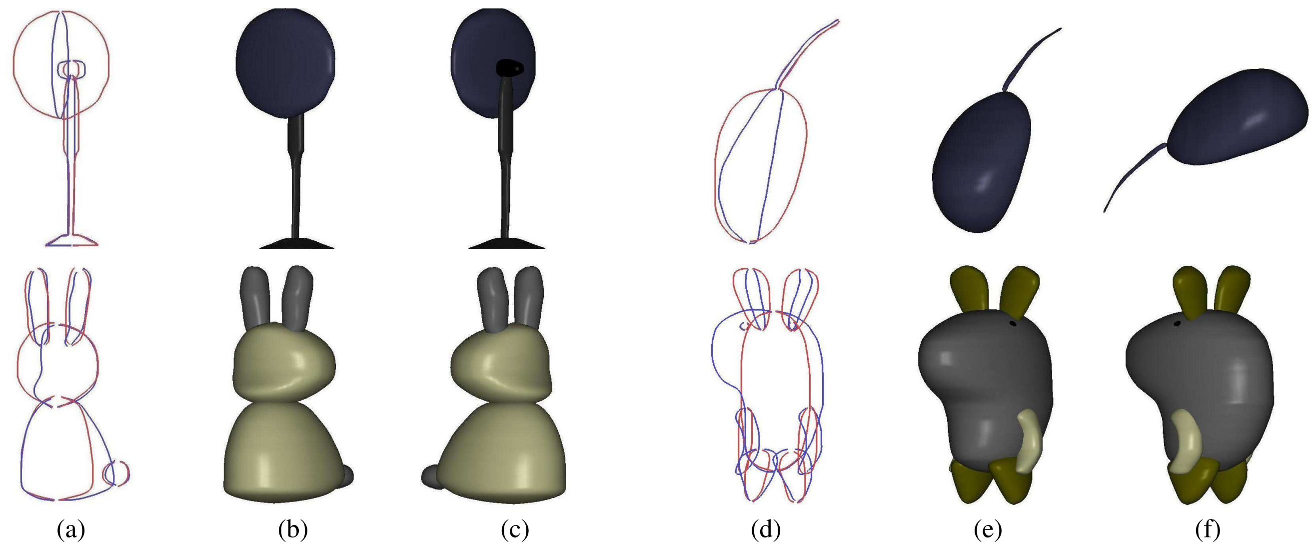


Fig. 8. *Creation of different free-form shapes from the user input sketches. The first figure of each group shows the user input 2D sketches (see columns (a,d)), and other figures of each group list the final generated 3D free-form shapes which are shown in two different view directions respectively (see columns (b,c) and columns (e,f)).*

The column (b) in Fig. 9 gives the user depicting sketches according to the referenced drawings, and the columns (c,d) in Fig. 9 list the final generated 3D free-form objects shown in two different view directions respectively. We can notice that some local sharp features may be generated by our modeling algorithm, such as the beak of the bird in the last row of Fig. 9. However, we can also notice that the reconstructed 3D shape may be a little thicker than the real object. This is because the constructive curves may not be the silhouettes of the reconstructed objects.

B. Comparisons with other 3D object reconstruction algorithm

Given a single view of 2D sketches, our reconstruction approach can generate complex free-form shapes effectively and conveniently. It should be pointed out that our reconstruction approach relies only on the 2D sketch content directly, and thus it can avoid some fuzzy and complicated operations to assemble multiple shape components. Moreover, the important advantage of our SymmSketch system is that it can provide more freedom for effectively controlling the final reconstructed shapes by employing four constructive curves. Compared with the most closely related approach presented by Cordier et al. [19], our 3D object reconstruction method can create complex free-form shapes with specific geometric features, such as the sharp jagged tail of women's hair clasp, the flat wine bottle, and the peaked cap as shown in Fig. 10.

C. Limitations of our algorithm

Our symmetric free-form object reconstruction approach always creates each shape component from four constructive curves, whether it is a self-symmetric shape component or a shape component symmetric to another one. For correctly recognizing and reconstructing 3D complex shapes, one limitation of our algorithm is that the input sketches should be drawn on a sketching plane according to a special order to infer the free-form shape (see Fig. 4). It should be mentioned that our modeling system only adopts one unique symmetry plane for creating the whole shape. Of course, it can generate some new free-form objects if introducing multiple symmetry planes. However, this potential scheme will increase some difficulties with the user inputting task for simultaneously handling several symmetry planes on one sketching plane, and may lead to some new user interactions for sketching different shape components.

Another limitation of our modeling method is that some fork-like shapes are difficult to be recovered correctly. As an example, Fig. 11 shows an unexpected reconstruction result of a funnel like shape using our method for a fork-like structure when the whole symmetric object is recovered as a single component. As a future work, some flexible mechanisms for generating such shapes should be investigated.

VI. Conclusions and Future Work

This paper presents a SymmSketch system for creating symmetric 3D free-form shapes which consist of two types of shape components, i.e. self-symmetric components and ones that are symmetric to another component with respect to a symmetry plane. Only with few strokes drawn by the user, our reconstruction method can automatically infer the relative depth of different shape components due to the mirror symmetry property. The experimental results illustrate its effectiveness for generating various types of free-form symmetric shapes.

Our future work may focus on the following issues. The current system will restrain the user from drawing the sketches in an arbitrary order. An algorithm for matching the symmetric curves automatically seems suitable to solve this issue. Moreover, as it is mentioned before, our modeling system can only reconstruct the free-form shape that is symmetric with respect to a unique symmetry plane. Another extension is to develop

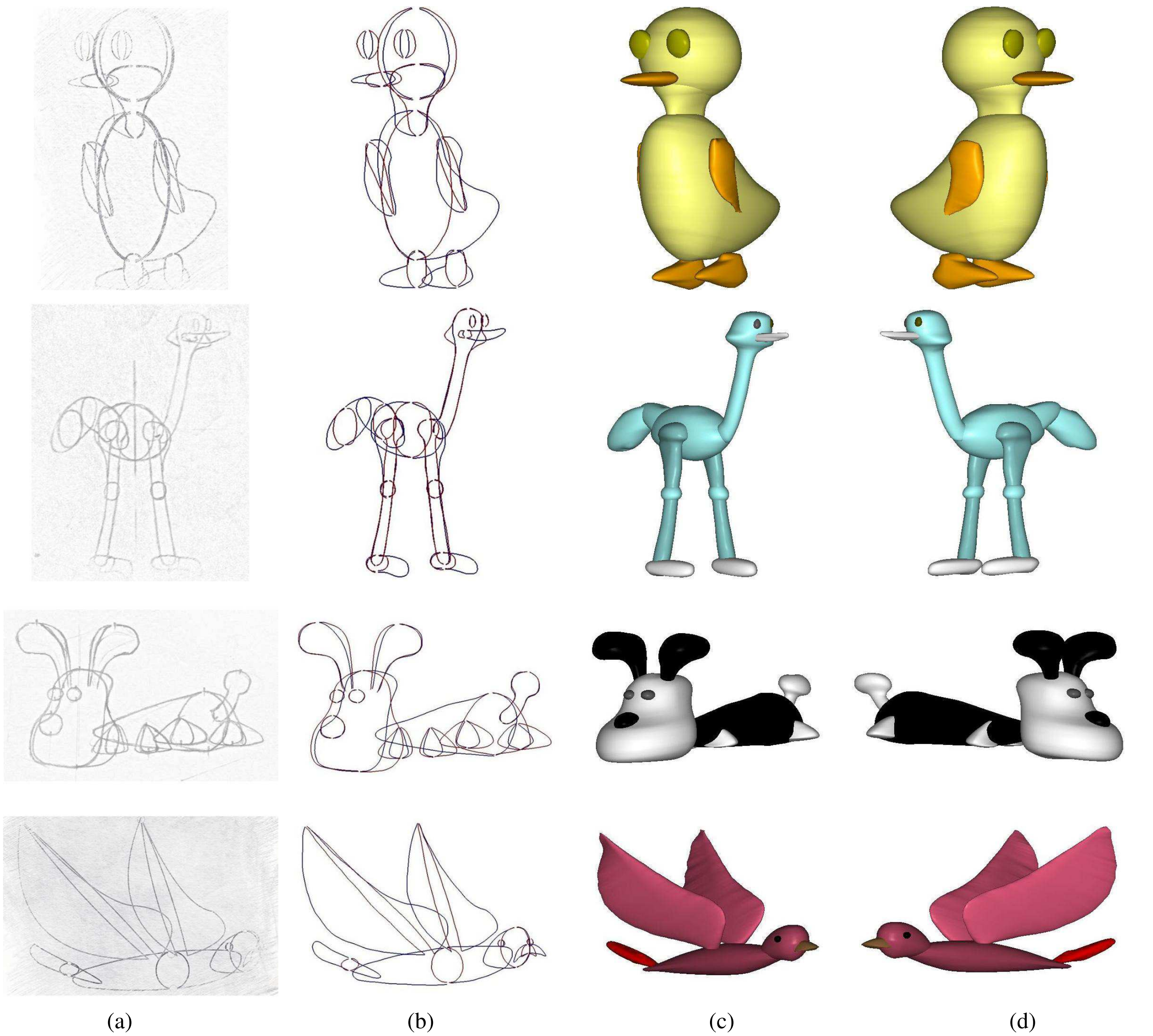


Fig. 9. *Different free-form shapes reconstructed by depicting the existing line drawings. (a) shows the existing line drawings; (b) gives the user input sketches; and (c,d) list the final generated symmetric 3D free-form shapes shown in two different view directions respectively.*

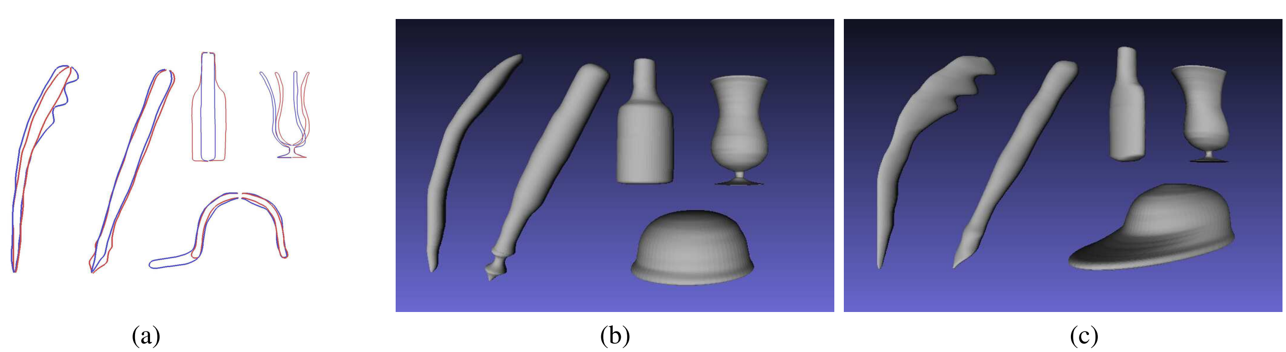
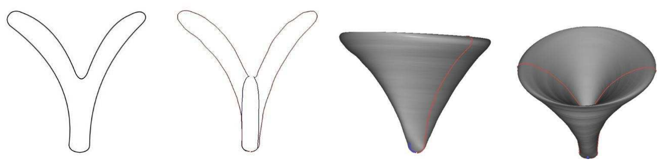


Fig. 10. *Comparison of the reconstructed free-form shapes by different methods. (a) gives the user input sketch containing the symmetric curves and the general curves; (b) shows the reconstructed 3D shapes only using two symmetric curves like Cordier et al.'s method [19]; and (c) lists the reconstructed 3D shapes using our proposed approach.*

a 3D modeling system for generating complex shapes which may consist of several local mirror symmetries with respect to their different symmetry planes.



(a) (b) (c) (d)

Fig. 11. *An unexpected reconstruction result of a funneled-like shape (c,d) using our method when creating a fork-like structure (a).*

References

[1] P. Blair. *Advanced Animation*. Walter T. Foster Publisher, 1949.

[2] L. Olsen, F. Samavati, M. Sousa, and J.-A. Jorge, "Sketch-based modeling: a survey", *Computers & Graphics*, vol. 33, no. 1, pp. 85-103, 2009.

[3] Y. Gingold, D. Zorin, and T. Igarashi, "Structured annotations for 2d-to-3d modeling", *ACM Transactions on Graphics*, vol. 28, no. 5, Article No. 148, 2009.

[4] A. Shtof, A. Agathos, Y. Gingold, A. Shamir, and D. Cohen-Or, "Geosemantic snapping for sketch-based modeling", *Computer Graphics Forum*, vol. 32, No. 2, pp. 245-253, 2013.

[5] P. Company, A. Piquer, M. Contero, and F. Naya, "A survey on geometrical reconstruction as a core technology to sketch-based modelling', *Computers & Graphics*, vol. 29, No. 6, pp. 892-904, 2005.

[6] R. Zeleznik, K. Herndon, and J. Hughes, "SKETCH: an interface for sketching 3d scenes", *Proc. of ACM SIGGRAPH*, 1996, pp. 163-170.

[7] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: a sketching interface for 3d freeform design", *Proc. of ACM SIGGRAPH*, 1999, pp. 409-416.

[8] R. Schmidt, B. Wyvill, M. Sousa, and J.-A. Jorge, "ShapeShop: sketch-based solid modeling with blobtrees", *Proc. of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2005, pp. 53-62.

[9] O. Karpenko and J. Hughes, "Smoothsketch: 3d free-form shapes from complex sketches", *ACM Transactions on Graphics*, vol. 25, No. 3, pp. 589-598, 2006.

[10] A. Nealen, T. Igarashi, O. Sorkin, and M. Alexa, "FiberMesh: designing freeform surfaces with 3d curves", *ACM Transactions on Graphics*, vol. 26, No. 3, Articel No. 41, 2007.

[11] S.-H. Bae, R. Balakrishnan, and K. Singh, "ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models", *ACM Symposium on User Interface Software and Technology*, 2008, pp. 151-160.

[12] P. Borosán, M. Jin, D. DeCarlo, Y. Gingold, and A. Nealen, "Rigmesh: automatic rigging for part-based shape modeling and deformation", *ACM Transactions on Graphics*, vol. 31, No. 6, Articel No. 198, 2012.

[13] K. Xu, K. Chen, H. Fu, W.-L. Sun, and S.-M. Hu, "Sketch2Scene: sketch-based co-retrieval and co-placement of 3d models", *ACM Transactions on Graphics*, Vol. 32, No. 4, Article No. 123, 2013.

[14] Z. Levi and C. Gotsman, "ArtiSketch: a system for articulated sketch modeling", *Computer Graphics Forum*, Vol. 32, No. 2, pp. 235-244, 2013.

[15] B. Seok-hyung and K. Won-sup, "Digital styling for designers: 3d plane-symmetric freeform curve creation using sketch interface", *Lecture Notes in Computer Science*, vol. 2669, pp. 701-710, 2003.

[16] A. Piquer, R.R. Martin, and P. Company, "Skewed mirror symmetry for depth estimation in 3d line-drawings", *Lecture Notes in Computer Science*, vol. 3088, pp. 142-153, 2004.

[17] X. Chen, S.-B. Kang, Y. Xu, J. Dorsey, and H.-Y. Shum, "Sketching reality: realistic interpretation of architectural designs", *ACM Transactions on Graphics*, vol. 27, No. 2, Article No. 11, 2008.

[18] S.-U. Cheon and S. Han, "A template-based reconstruction of plane-symmetric 3d models from freehand sketches", *Computer-Aided Design*, Vol. 40, No. 9, pp. 975-986, 2008.

[19] F. Cordier, H. Seo, J. Park, and J.-Y. Noh, "Sketching of mirror-symmetric shapes", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 11, pp. 1650-1662, 2011.

[20] A. Andre and S. Saito, "Single-view sketch based modeling", *Proc. of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, 2011, pp. 133-140.

[21] J. Mitani, H. Suzuki, and F. Kimura, "3D sketch: sketch-based model reconstruction and rendering", *The 7th Workshop on Geometric Modeling: Fundamentals and Applications*, 2000, pp. 85-98.

[22] P. Varley, Y. Takahashi, J. Mitani, and H. Suzuki, "A two-stage approach for interpreting line drawings of curved objects", *Proc. of the 1st Eurographics Symposium Sketch-Based Interfaces and Modeling*, 2004, pp. 117-126.

[23] A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or, "A sketch-based interface for detail-preserving Mesh editing", *ACM Transactions on Graphics*, Vol. 24, No. 3, pp. 1142-1147, 2005.

[24] M. Masry and H. Lipson, "A sketch-based interface for iterative design and analysis of 3d objects", *ACM Transactions on Graphics*, Vol. 26, No. 3, Article No. 31, 2007.

[25] Y. Li, Z. Pizlo, and R.-M. Steinman, "A computational model that recovers the 3d shape of an object from a single 2d retinal representation", *Vision research*, Vol. 49, No. 9, pp. 979-991, 2009.

[26] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for 3d models", *ACM Transactions on Graphics*, Vol. 22, No. 1, pp. 83-105, 2003.

[27] H. Shin and T. Igarashi, "Magic canvas: interactive design of a 3d scene prototype from freehand sketches", *Proc. of the Graphics Interface*, 2007, pp. 63-70.

[28] J. Lee and T. Funkhouser, "Sketch-based search and composition of 3d models", *Proc. of the 5th Eurographics conference on Sketch-Based Interfaces and Modeling*, 2008, pp. 97-104.

[29] Z. Li, J. Liu, and X. Tang, "A closed-form solution to 3d reconstruction of piecewise planar objects from single images", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1-6.

[30] Y. Wang, Y. Chen, J. Liu, and X. Tang, "3D reconstruction of curved objects from single 2d line drawings", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1834-1841.

[31] T. Xue, J. Liu, and X. Tang, "3-D modeling from a single view of a symmetric object", *IEEE Transactions on Image Processing*, Vol. 21, No. 9, pp. 4180-4189, 2012.

[32] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: extracting editable objects from a single photo", *ACM Transactions on Graphics*, Vol. 32, No. 6, Article No. 195, 2013.

[33] A.-C. Öztireli, U. Uyumaz, T. Popa, A. Sheffer, and M. Gross, "3D modeling with a symmetric sketch", *Proc. of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, 2011, pp. 23-30.

[34] F. Cordier, H. Seo, M. Melkemi, and N.-S. Sapidis, "Inferring mirror symmetric 3d shapes from sketches", *Computer-Aided Design*, Vol. 45, No. 2, pp. 301-311, 2013.

[35] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide (5th Edition)*. Morgan Kaufmann Publishers, 2002.

[36] A. Severn, F. Samavati, J. Cherlin, M. Sousa, and J.-A. Jorge, "Sketch-based modeling and assembling with few strokes", *Proc. of the 8th Eurographics Symposium Sketch-Based Interfaces and Modeling*, 2011, pp. 255-286.

Yongwei Miao received his Ph.D. in computer graphics from the State Key Lab. of CAD & CG at Zhejiang University in March 2007. From February of 2008 to February of 2009, he worked as a visiting scholar in University of Zürich, Switzerland. From November of 2011 to May of 2012, he worked as a visiting scholar in University of Maryland, USA. Dr. Miao is a professor in the College of Computer Science and Technology, Zhejiang University of Technology, P. R. China. His research interests include computer graphics, digital geometry processing, visual media computing, 3D reconstruction and computer vision.

Feixia Hu is a master candidate in the College of Computer Science and Technology, Zhejiang University of Technology, P.R. China. His research interests include computer graphics, visual media computing, and 3D reconstruction.

Xudong Zhang is a faculty in the College of Computer Science and Technology, Zhejiang University of Technology, P.R. China. His research interests include computer graphics, visual media computing, and computer vision.

Jiazhou Chen is a lecturer in the College of Computer Science and Technology, Zhejiang University of Technology, P. R. China. Before that, he was a joint Ph.D. candidate between Bordeaux University (INRIA Bordeaux Sud-Ouest), France and Zhejiang University (the State Key Lab. of CAD & CG at Zhejiang University), China. He obtained the French doctoral diploma from Bordeaux University in July 2012, and also obtained the Chinese doctoral diploma from Zhejiang University in December 2012. His research interests include computer graphics, image and video stylization, augmented reality, and visual media computing.

Renato Pajarola received a Dr. Sc. Techn. in Computer Science in 1998 from the Swiss Federal Institute of Technology (ETH) Zurich. After a postdoc in the Graphics, Visualization & Usability Center at Georgia Tech., he joined the University of California Irvine in 1999 as an Assistant Professor where he founded the Computer Graphics Lab. He is a now Full Professor in the Department of Informatics, University of Zurich, Switzerland. His research interests include real-time 3D graphics, scientific visualization and interactive 3D multimedia.